ELSEVIER

Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Economics and Finance 3 (2012) 852 – 857

Procedia
Economics and Finance

www.elsevier.com/locate/procedia

Emerging Market Queries in Finance and Business

New approach on optimal decision making based on formal automata models

Avram Călin[a,*], Gligor Adrian[b], Avram Eleonora Laura[a]

[a]*University Dimitrie Cantemir from Tirgu-Mures, Bodoni Sandor st., 3-5, Tirgu-Mures, Romania*
[b]*University Petru Maior from Tirgu-Mures, Nicolae I. st., 1 Tirgu-Mures, Romania*

Abstract

The main idea of paper is to present a new proposal for formal languages useful in modeling of many type of processes such as: economic, computing, social, biological, etc. Our approach is developed to describe computational models that can be implemented using established concepts such as different implementations of automata, etc.. The paper describes how is defined the new language based on developed rules and its grammar, upon which is build a computational model for optimal decision making in case of a process control. Verification and validatation of the proposed theory is supported by an experimental implementation that aims to optimize a generic flow in a economic process based on automaton described by proposed formal language.

Keywords: formal language, automaton, process control, optimization, economic proces

* Corresponding author.
E-mail address: avram_calin@yahoo.com.

doi:10.1016/S2212-5671(12)00240-7

1. Introduction

The main idea of paper is to present an useful formal language in modeling different economic processes, of software development, social development, etc. This language is conceived to describe computational models that can be implemented using established concepts such as different implementations of automaton. In this paper we described how the new language and its grammar were defined, on which we tried to build a computer model to ensure the optimal decision support.

To verify and validate the proposed theory was used an experimental implementation based on a proposed formal language applying an automaton. The chosen process pursued the flow optimization in an economic process.

2. Problem Formulation

The notion of *formal and abstract* received new dimensions once with interdisciplinary research of applied mathematics in literature, art, history, biology and of course in economics. Through a formal language is defined an assemblage of strings elements from an alphabet. These strings are called well-formed words or sentences or grammatically correct sentences. These words in a language are not chosen at random, they must comply with the rules and the set of these rules constitutes the *formal grammar*.1, 2, 4.

To find out which is the directed graph grammar that is intended to calculate the maximum flow, we assume the following case study. In this case the graph that was chosen has a large number of nodes that by solving the problem with Ford-Fulkerson algorithm results an easy method of solving, even if the graph has a large number of nodes.

A transport company has 35 trucks that wants to move from point A to point J. Moving the 35 trucks from one place to another is done in stages, so that in the first stage as many of them must reach the point J; on their way, the trucks should make some break in other intermediary points B, C, D, E, F, G, H, I, J. Conditions for the reception, supply etc., lead to the existence of a limitation of the routes; the existing capacities are listed on the network arcs.3,6.

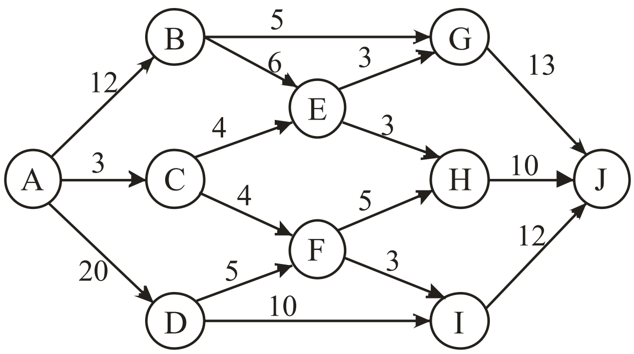

Fig. 1 The problem with Ford-Fulkerson algorithm

Determine an optimal transportation plan so that, at this stage may leave as many trucks to the point J.

3. Problem Solution

Maximum flow problem that crosses the transportation network has for solving the following mathematical form using linear programming 6, 7:

$$\begin{cases} z_{\max} = \varphi_{\max} \quad under \\ 0 \le \varphi'_{ij} \le c'_{ij} \quad (i, j = \overline{0, n+1}); \\ \sum_{j=0}^{n+1} \varphi'_{ij} = \sum_{j=0}^{n+1} \varphi'_{ij} \quad (i = \overline{1, n}); \\ \sum_{j=0}^{m+1} \varphi'_{j0} = \sum_{j=0}^{n+1} \varphi'_{0j} - \varphi \\ \sum_{j=0}^{n+1} \varphi'_{j,n+1} = \sum_{j=0}^{n+1} \varphi'_{n+1} + \varphi \end{cases} \tag{1}$$

where $u = (x_i, y_j)$ is the arc, φ is the flow value and $\varphi'_{(u)} = \varphi'_{ij}$ is the arc flow.

The automaton that corresponds the algorithm is $M = (Q, \Sigma, \delta, q_0, F)$ has the following value: $Q = \{A, B, C, D, E, F, G, H, I, J\}$, $F = \{J\}$, the input alphabet is given by $\Sigma = \{12,3,20,5,6,4,4,5,10,3,3,5,3,13,10,12\}$, the transition functions are defined as follows:

$\delta(A,12) = B$ $\quad$ $\delta(B,5) = G$ $\quad$ $\delta(C,4) = E$

$\delta(A,3) = C$ $\quad$ $\delta(B,6) = E$ $\quad$ $\delta(C,4) = F$

$\delta(A,20) = D$

$\delta(D,5) = F$ $\quad$ $\delta(E,3) = G$ $\quad$ $\delta(F,5) = H$

$\delta(D,10) = I$ $\quad$ $\delta(E,3) = H$ $\quad$ $\delta(F,3) = I$

$\delta(G,13) = J$ $\quad$ $\delta(H,10) = J$ $\quad$ $\delta(I,12) = J$

The regular grammar $G = (V_N, V_T, S, P)$ where $V_N = \{A, B, C, D, E, F, G, H, I, J\}$, $V_T = \{12,3,20,5,6,4,4,5,10,3,3,5,3,13,10,12\}$ with the set of rules for generating:

1) $A \to 12B$ $\quad$ 2) $A \to 3C$ $\quad$ 3) $A \to 20D$

4) $B \to 6E$ $\quad$ 5) $B \to 5G$

6) $C \to 4E$ $\quad$ 7) $C \to 4F$

8) $D \to 5F$ $\quad$ 9) $D \to 10I$

10) $E \to 3G$ $\quad$ 11) $E \to 3H$

12) $F \to 5H$ $\quad$ 13) $F \to 3I$

14) $G \to 13J$

15) $H \to 10J$

16) $I \to 12J$

17) $J \to \lambda$

Possible derivations from the grammar are:

$$A \to^{(1)} 12 \cdot B \to^{(5)} 12 \cdot 5G \to 12 \cdot 5 \cdot 13J \to^{(17)} 12 \cdot 5 \cdot 13$$
$$A \to^{(1)} 12B \to^{(4)} 12 \cdot 6E \to^{(10)} 12 \cdot 6 \cdot 3G \to^{(14)} 12 \cdot 6 \cdot 3 \cdot 13J \to^{(17)} 12 \cdot 6 \cdot 3 \cdot 13$$
$$A \to^{(1)} 12B \to^{(4)} 12 \cdot 6E \to^{(11)} 12 \cdot 6 \cdot 3H \to^{(15)} 12 \cdot 6 \cdot 3 \cdot 10J \to^{(17)} 12 \cdot 6 \cdot 3 \cdot 10$$
$$A \to^{(2)} 3C \to^{(6)} 3 \cdot 4E \to^{(10)} 3 \cdot 4 \cdot 3G \to^{(14)} 3 \cdot 4 \cdot 3 \cdot 13J \to^{(17)} 3 \cdot 4 \cdot 3 \cdot 13$$
$$A \to^{(2)} 3C \to^{(6)} 3 \cdot 4E \to^{(11)} 3 \cdot 4 \cdot 3H \to^{(15)} 3 \cdot 4 \cdot 3 \cdot 10J \to^{(17)} 3 \cdot 4 \cdot 3 \cdot 10$$
$$A \to^{(2)} 3C \to^{(7)} 3 \cdot 4F \to^{(12)} 3 \cdot 4 \cdot 5H \to^{(15)} 3 \cdot 4 \cdot 5 \cdot 10J \to^{(17)} 3 \cdot 4 \cdot 5 \cdot 10$$
$$A \to^{(2)} 3C \to^{(7)} 3 \cdot 4F \to^{(13)} 3 \cdot 4 \cdot 3I \to^{(16)} 3 \cdot 4 \cdot 3 \cdot 12J \to^{(17)} 3 \cdot 4 \cdot 3 \cdot 12$$
$$A \to^{(3)} 20D \to^{(8)} 20 \cdot 5F \to^{(12)} 20 \cdot 5 \cdot 5H \to^{(15)} 20 \cdot 5 \cdot 5 \cdot 10J \to^{(17)} 20 \cdot 5 \cdot 5 \cdot 10$$
$$A \to^{(3)} 20D \to^{(8)} 20 \cdot 5F \to^{(13)} 20 \cdot 5 \cdot 3I \to^{(16)} 20 \cdot 5 \cdot 3 \cdot 12J \to^{(17)} 20 \cdot 5 \cdot 3 \cdot 12$$
$$A \to^{(3)} 20D \to^{(9)} 20 \cdot 10I \to^{(16)} 20 \cdot 10 \cdot 12J \to^{(17)} 20 \cdot 10 \cdot 12$$

The generated language is L={ $12 \cdot 5 \cdot 13$, $12 \cdot 6 \cdot 3 \cdot 13$, $12 \cdot 6 \cdot 3 \cdot 10$, $3 \cdot 4 \cdot 3 \cdot 13$, $3 \cdot 4 \cdot 3 \cdot 10$, $3 \cdot 4 \cdot 5 \cdot 10$, $3 \cdot 4 \cdot 3 \cdot 12$, $20 \cdot 5 \cdot 5 \cdot 10$, $20 \cdot 5 \cdot 3 \cdot 12$, $20 \cdot 10 \cdot 12$ }

In case that for each defined sub graph is established the maximum flow, this is of 41 trucks which starting from point A reach the point J.. This is not correct because there are duplicates of the minimum flows on certain routes which must be eliminated. Solving the problem using the Ford-Fulkerson algorithm, the maximum flow that is starting from A and is reaching the point J is 28 trucks.

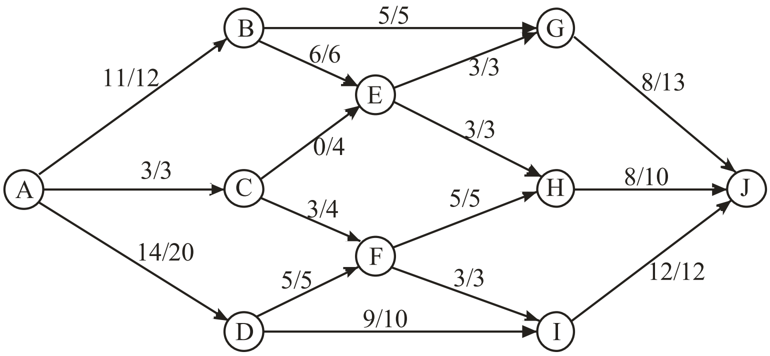

Fig. 2 Solving the problem with Ford-Fulkerson algorithm

Can see that the difference between the two values is on the transport capacity of the G, H and I points. These values are:

- for the G point, the capacity that can support is 8 and from grammar we get 11;
- for the H point, the capacity that can support is 8 from the grammar we get 14;
- for the I point the capacity that can support is 12 from the grammar we get 16.

The problem that arises is for I point that through the grammar obtained above we have the values given by subgraphs: ACFIJ, ADFIJ and ADIJ with the minimum capacity of 3, 3:10. Point I may have a maximum flow as solving the problem, using the algorithm we have only 12 which follows that a subgraph of two ACFIJ, ADFIJ actually have the minimum capacity of 2 and the other is doubled as a value.

To remove the duplicate values we didn't find any method to bring a plus and lead to an equivalent result to that is found by the formalism which uses the Ford-Fulkerson algorithm.

The pseudo code for the application of this algorithm is the following:

- Let *VN* be the alphabet of the model
- Let *VT* be a set of rules of the model
- Set *W* based on *VN* and *VT*
- Set *L* to a empty matrix of size of *W*

- Set *LR* to a empty matrix of size of *W*
- Set *grammar* to an empty colection
- Compute the matrix *L* based on *VN*
- Compute the matrix *L* based on *VN*
- Compute *maxWordLength*
- Set *wordLength* to 1
- For each *wordLength* from available grammar
- Set *LL* to a empty matrix of size of *W*
- Compute latin product between *L* and *LR*
- Set *LL* to *L*
- For each *element* of *L*
- Extract *word* of length *wordLength* in *grammar*
- End For
- End For
- For each *word* from *grammar*
- Compute the capacity of the *word* in *grammar*
- End For
- Sort *grammar* on *wordLenght*
- Set *GR* to an empty colection
- Set *wordLength* to 2
- For each *wordLength* from *grammar*
- Search and extract word with minimum capacity in *GR*
- End For
- Save *GR*

4. Conclusion

In this paper is proposed the modeling of certain classes of economic processes starting from a formalism based on a formal language. The paper describes how was defined the new language and its grammar, on which to build a computer model to ensure the optimal decision support. To verify and validate the proposed theory was conceived a case study for optimizing the flow in an economic process where is intended to find the maximum flow in a network transport.

For modeling of processes in general, or the economic ones in particular, associated implementation models as directed graphs in formal languages can be achieved and implemented only if the graph has a limited number of nodes. If the graph has a large number of nodes, formalizing and obtaining its automaton involves a large number of operations and the optimization of this method is becoming increasingly complex.

As a result of a detailed study and analysis it is found that the results provided by formalism which implements the Ford-Fulkerson algorithm are suitable in the case of solving certain decision problems associated with transport economic processes and not only.

Appealing to formal languages can lead to build computational models that can be used in modeling of large class of economic processes and for which optimal solutions can be found appealing at appropriate computer implementations that can benefit of new or consecrated algorithms.

References:

Atanasiu A, *Limbaje formale si automate*, Editura Infodata, 2007.
Atanasiu I., Raicu D., Sion R., Mocanii I., *Limbaje Formale si Automate*, Îndrumar pentru aplicații, 2002.
Cormen H. Thomas, Charles E. Leiserson, Rivest R. Roland, *Introducere în algoritmi*, Editura Agora,2002.
Jalobeanu C, Marinescu D, *Bazele teoriei calculului limbaje formale si automate*, Editura Albastră, 2006;
Paun Gh., *Mecanisme generative ale proceselor economice*, Editura Tehnică, 1980.
Rădescu N., Rădescu E., *Probleme de teoria grafurilor*, Editura Scrisul Românesc, 1982.
Minuț P., Matematici aplicate în economie, Târgu Mures, Editura „Dimitrie Cantemir”, 2002